

一. 名词解释

1. 地址映射: 将程序中的逻辑地址转换为内存中的物理地址的过程。
2. 动态重定位: 在程序运行时, 将逻辑地址转换为物理地址的技术, 允许程序加载到内存任意位置。
3. 虚拟内存: 通过请求调页和页面置换技术, 将磁盘空间作为内存扩展, 为用户提供比实际物理内存更大的逻辑地址空间。
4. 静态链接: 程序运行前, 将所有模块及库函数链接成单一可执行文件。
5. 对换: 将暂时不运行的进程从内存移至磁盘(换出), 再将需要执行的进程从磁盘调入内存的内存管理技术。
6. 设备驱动程序: 操作系统内核中管理特定硬件设备的软件模块, 为上层提供统一接口, 控制设备操作。
7. Spooling: 通过磁盘缓冲区模拟独占设备的共享操作, 将低速 I/O 设备虚拟为高速设备。
8. I/O 通道: 一种专用处理器, 可独立控制 I/O 操作, 减少 CPU 干扰, 提高并行性。
9. 文件系统: 操作系统用于组织、存储和访问文件的一种数据结构与算法集合。
10. 目标文件: 由编译器生成, 包含机器代码但未链接的中间文件, 含重定位信息。
11. 文件的逻辑结构: 用户视角下的文件组织方式, 如流式、记录式结构。
12. 有符号文件: 由逻辑记录组成的文件, 支持按记录关键字访问。
13. 位示图: 一种利用二进制位, 表示磁盘块/内存页使用状态的存储管理数据结构。
14. 程序接口: 操作系统提供给应用程序的编程接口, 用于调用系统功能。
15. 系统调用: 操作系统内核为应用程序提供的底层服务接口, 需通过中断或陷阱指令触发。
16. I/O 中断: 当 I/O 操作完成或出错时, 设备控制器向 CPU 发送的硬件信号, 触发中断处理程序。
17. 文件管理系统: 操作系统组件集合, 负责文件的创建、读写、权限控制及存储空间管理。
18. 文件: 具有符号名的相关数据集合, 存储在物理设备上。
19. 文件的逻辑结构: 从用户角度看到的文件组织形式。
20. 文件的物理结构: 文件在存储设备上的实际分配方式, 如连续分配、链式、索引分配。

二.

1. 地址重定位
2. 绝对装入, 可重定位装入, 动态运行时装入
3. 静态链接, 临时动态链接, 运行时动态链接
4. 空闲链表, 空闲位图
5. 紧凑
6. 固定, 系统, 可变, 程序
7. 对换性, 虚拟性
8. 文件管理, 目录管理, 磁盘管理
9. 空闲分区管理, 分区分配算法, 碎片处理
10. 内部碎片和外部碎片
11. 页, 分段, 段页式
12. 快表
13. 逻辑地址空间, 物理内存
14. 逻辑独立性, 易于共享和保护, 支持动态增长, 便于动态链接, 消除内部碎片
15. 内存管理单元
16. 局部性原理 时间局部性, 空间局部性。

17. 处理机 指令类型单一, 与CPU共享内存.
18. 串行文件, 索引文件, 索引/索引文件
19. 单级索引, 多级索引, 增量式索引
20. 动态重定位
21. 用户层I/O接口, 设备无关层, 设备驱动程序
22. 连续分配, 链接分配, 索引分配, 索引/索引
23. 主存, 辅存(磁盘), cache
24. 连续分配, 离散分配
25. 存储设备特性
26. 无结构文件, 有结构文件, 树形文件
27. 连续分配, 链接分配, 索引分配
28. 绝对路径, 相对路径
29. 隐式链接, 显式链接
30. 硬链接, 软链接
31. 块设备, 字符设备, 独占设备, 共享设备, 虚拟设备
32. DMA方式, 通道方式
33. 单缓冲, 循环缓冲, 缓冲池
34. 寻道时间, 传输时间

三.

1. 寄存器 → cache → 内存 → 外存
性能从上到下 ↓, 成本由低到高 ↑, 容量从上到下 ↑
2. ① 编写: 用高级语言、汇编编写源代码
② 编译: 编译器将源代码转换为目标代码
③ 链接: 链接器将目标代码与库函数合并成可执行文件 .exe
④ 装入: 加载器将可执行文件装入内存, 分配资源
⑤ 运行: CPU执行指令, 操作系统管理进程调度, 内存分页等.
3. ① 静态链接, 动态链接, 运行时链接
② 同: 都是为了让程序调用外部库代码最终生成可执行文件的进程.
异: 静态链接是在编译时将所有用到的库代码嵌入到可执行文件, 而动态链接是目标模块装入内存时边装入边链接, 运行时链接是程序执行中需要目标模块时再对应进行链接.
4. ① 首次适应: 从空闲分区的起始位置顺序查找, 找到第一个满足要求的就分配
② 最佳适应: 总是选择容量最小且满足需求的空闲分区分配.
③ 最坏适应: 总是选择容量最大的空闲分区进行分配.
④ 邻近适应: 类似首次适应, 但扫描起点是上次分配的分区的后.

5. 动态分配: 首次适应, 最佳适应, 最坏适应, 邻近适应.

CPU调度: 先来先服务, 短作业优先, 时间片轮转, 高响应比优先.

同: 均需选择资源分配策略, 追求提高资源利用率.

异: 内存算法目标减少内存碎片, 操作对象是应用分区, 预测提高内存分区利用率.

CPU算法目标是减少平均等待时间, 使得对CPU利用更高效, 操作对象为就绪进程队列.

6. 页面置换: 最佳置换, 先进先出, 最近最久未使用

CPU调度: 先来先服务, 短作业优先, 时间片轮转, 高响应比优先.

同: 均采用队列管理策略, 依据历史行为做预测.

异: 页面置换目标降低缺页率, 操作对象为磁盘等外存.

CPU调度算法目标: 提高CPU利用率, 操作对象为就绪进程队列.

7. 页面置换: 最佳置换, 先进先出, 最近最久未使用

磁盘调度: 先来先服务, 最短寻道优先, 电梯算法

同: 均需选择服务顺序, 减少等待时间.

异: 磁盘调度使磁头移动轨迹, 页面置换无物理移动

8. 分页存储: 根据逻辑地址计算出块号 + 页内偏移, 通过页表查得物理块号, 物理地址 = 物理块号 + 页内偏移.

分段存储: 根据逻辑地址计算出段号 + 段内偏移, 查段表得到段基址.

物理地址 = 段基址 + 偏移

段式存储: 根据逻辑地址计算出段号 + 段内偏移, 查段表找到页基址.

页表号查页表得到物理块号. 物理地址 = 块号 + 页内偏移.

9. ①关中断 → ②保存现场 → ③识别中断源 → ④执行中断服务程序 → ⑤恢复现场 → ⑥开中断返回.

10. 用途: 系统文件, 用户文件, 库文件

数据模型: 源文件, 目标文件, 可执行文件.

组织方式: 普通文件, 目录文件, 特殊文件.

管理方式: 只读, 读写, 只执行.

11. 单级目录: 所有文件放同一目录 → 遍历查找

两级目录: 用户目录 + 用户文件, 先查用户目录再查文件.

树形目录: 层次化路径 → 路径解析逐层查询

12. 顺序访问, 索引访问, 顺序索引访问.

顺序访问: 批量访问快但随机访问慢, 且增删困难. 索引访问: 交互高支持随机访问, 增删更简单. 但记录多时, 索引会占用很多空间, 顺序索引访问解决了前两个缺点.

3. 内存分配方式: 连续分配, 分页分配, 分段分配

外存分配方式: 连续分配, 链式分配, 索引分配.

同: 二者目标都是为了减少碎片, 更充分利用空间

异: 内存分配连续分配, 外存连续分配或索引分配.

4. ①采用高效调度算法 ②磁盘缓存: 在内存中保留磁盘块副本 ③延迟写入, 批量写入磁盘.